

## 1.8. АНАЛИЗ СРЕДСТВ КОНТРОЛЯ ПОВЫШЕНИЯ ЭНЕРГОЭФФЕКТИВНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Егурнов В.А., Абдрахманов Д.Л. ВолгГТУ, г. Волгоград

*В настоящее время повышение энергоэффективности программного обеспечения и вычислительных систем в целом становится как никогда актуальным. В работе были исследованы методы и средства контроля и повышения энергоэффективности программного обеспечения. Рассмотрены средства мониторинга потребления микропроцессорами энергетических ресурсов, исследованы области применения и ограничения данных приложений. Также был рассмотрен интерфейс RAPL, разработанный компанией Intel для контроля и ограничения энергопотребления микропроцессоров и некоторых подсистем вычислительной системы.*

### **Введение**

В современном мире вопрос энергоэффективности различных классов и архитектур вычислительных систем является чрезвычайно важным. В связи с удорожанием электроэнергии и введением все более жестких энергетических и экологических стандартов, наблюдаемых в целом ряде стран, уровень энергопотребления становится одним из немаловажных факторов при покупке различной аппаратуры. Все это в равной степени относится и к средствам вычислительной техники, от мобильных телефонов [Fallis, 2018] и небольших встраиваемых систем [Lutui, 2018], до вычислительных кластеров [Schill, 2017].

Помимо характеристик вычислительного оборудования, важным фактором является энергоэффективность программного обеспечения. При этом данная характеристика все чаще признается в качестве одного из важнейших атрибутов качества программного обеспечения, особенно для встроено-го программного обеспечения [Kim, 2018]. Повышенное энергопотребление влечет за собой ряд негативных последствий – от сокращения времени автономной работы для различных устройств до значительных счетов за потребленную электроэнергию в случае вычислительных кластеров. Повышенное энергопотребление приводит к увеличению тепловыделения и, как следствие, к увеличению затрат на охлаждение оборудования.

Для оценки энергоэффективности программного обеспечения используются различные методы. Так, в работе [Bán, 2019] описаны модели, позволяющие прогнозировать энергоэффективность программного обеспечения, которые используют как статические показатели исходного кода, так и динамические измерения времени выполнения, мощности и энергопотребления. Авторы исследования заявляют, что после обучения моделей можно использовать только статические показатели для прогнозирования. В работе [Kiselev, 2020] описана методика оценки энергоэффективности программного обеспечения на основе профилирования. Стоит отметить, что существуют и другие методики оценки и прогнозирования энергоэффективности программного обеспечения.

Для повышения энергоэффективности исследователи предлагают различные оптимизации программного обеспечения, связанные с использованием авторских оптимизаций [Kravets, 2022], оптимизирующих возможностей компиляторов [Guzma, 2013], другие оптимизации.

Большое количество работ посвящено проблемам снижения энергопотребления вычислительных кластеров, обработке больших данных [Majeed, 2015], использованию облачных технологий [Katal, 2022]. При этом разрабатываются методы для динамического управления узлами с целью снижения энергопотребления, а также стратегии контроля состояния узлов. Так, в работе [Ivannikov, 2010] рассмотрены методы снижения энергопотребления единого компьютерного кластера за счет гибких стратегий управления состояниями узлов (их пробуждением или выключением) и порядком выполнения ожидающих задач.

### **Средства мониторинга.**

Одним из способов анализа энергоэффективности программного обеспечения является мониторинг уровня потребления процессором энергетических ресурсов [Lyakhomskii, 2013, Nouredine, 2015, Ilyas, 2022]. При этом стоит отметить, что данная задача не является приоритетной для операционной системы. Существующее положение обуславливает недостаточную распространённость соответствующих приложений с открытым исходным кодом.

Разработчики программного обеспечения данной целевой направленности акцентируют своё внимание на создании утилит, которые поддерживают непосредственное функционирование системы, игнорируя необходимость учёта потребления энергии конкретным ядром процессора при решении широкого спектра задач. Указанная проблема осложняется отсутствием единого стандарта, регламентирующего с заданной точностью текущее потребление отдельного ядра для процессоров различных архитектур или для процессоров разных поколений с единой архитектурой.

В операционных системах на основе ядра Linux (как и в BSD-подобных системах) существует множество готовых решений, которые могут предоставить пользователю детализированную информацию о сенсорах, установленных в системе, как в формате «сырых данных» («raw data»), так и в виде интерпретации, адаптированной для пользователя. Основными приложениями для отслеживания текущего состояния процессора, характерными для таких операционных систем, являлись утилиты `top/ps`, которые были разработаны при реализации проекта GNU. Интерфейс одного из таких приложений представлен на рисунке 1.

```
top - 15:07:40 up 1:18, 0 users, load average: 0.02, 0.01, 0.00
Tasks: 5 total, 1 running, 4 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni,100.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
MiB Mem : 31628.9 total, 31110.0 free, 136.0 used, 382.9 buff/cache
MiB Swap: 8192.0 total, 8192.0 free, 0.0 used. 31128.5 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
1	root	20	0	896	532	464	S	0.0	0.0	0:00.01	init
921	root	20	0	896	88	20	S	0.0	0.0	0:00.00	init
922	root	20	0	896	88	20	S	0.0	0.0	0:00.00	init
923	citrull+	20	0	10040	5000	3300	S	0.0	0.0	0:00.02	bash
936	citrull+	20	0	9104	3428	2924	R	0.0	0.0	0:00.00	top

Рисунок 1 – Интерфейс утилиты `top`

Цель данных приложений состояла в предоставлении пользователю информации о состоянии задач, а также данных о текущей загрузке центрального процессора, объёме используемой памяти. При этом утилиты не позволяли получить сведения о потреблении процессором электроэнергии и температуре его ядер.

Совершенствование компонентной базы материнских плат и интеграция микросхем аппаратного мониторинга в начале 2000-х годов явились определяющими условиями создания утилиты `lm-sensors` [`Lm_sensors`]. Основная задача `lm-sensors` – получение данных чипов, предназначенных для регистрации следующих параметров: температура ядер процессора, скорость вращения вентилятора, уровень

```
(base) citrullux@headnode:~$ sensors
k10temp-pci-00c3
Adapter: PCI adapter
temp1: +23.9°C (high = +70.0°C)
          (crit = +70.0°C, hyst = +67.0°C)

k10temp-pci-00cb
Adapter: PCI adapter
temp1: +24.0°C (high = +70.0°C)

k10temp-pci-00d3
Adapter: PCI adapter
temp1: +20.5°C (high = +70.0°C)
          (crit = +70.0°C, hyst = +67.0°C)

k10temp-pci-00db
Adapter: PCI adapter
temp1: +20.8°C (high = +70.0°C)

fam15h_power-pci-00c4
Adapter: PCI adapter
power1: 64.01 W (crit = 114.64 W)

fam15h_power-pci-00d4
Adapter: PCI adapter
power1: 44.88 W (crit = 114.64 W)
```

Рисунок 2. Интерфейс `lm-sensors` отображающий данные доступных датчиков

`lm-sensors` самостоятельно не обращается к чипам мониторинга энергопотребления конкретного оборудования, работа утилиты реализуется через использование методов, реализованных в ядре операционной системы [`The Linux kernel user-space API guide`]. В связи с этим, использование `lm-sensors`

напряжения. Текстовый вариант интерфейса данного приложения приведён на рисунке 2.

В специальной литературе представлено подробное описание ряда микросхем, которые могут быть использованы при производстве материнских плат для регистрации актуальных для пользователя показателей. При этом существует класс вычислительных устройств, для которых реализация отслеживания необходимых данных затруднена, что обусловлено особенностями архитектуры. Например, устройство ноутбуков, в большинстве случаев, включает в своей структуре аппаратный чип мониторинга. В этом случае пользователю предоставляется опосредованный способ получения значения температур процессоров (BIOS или ACPI), пакет `lm-sensors` неприменим.

возможно только в том случае, если ядро операционной системы поддерживает передачу данных о текущем энергопотреблении. Еще одним недостатком lm-sensors является отсутствие прямых показаний потребляемой мощности, которая вычисляется с использованием информации о напряжении и силе тока, регистрируемых приложением.

Современные оболочки предоставляют пользователю интуитивно понятный интерфейс, интерпретирующий данные lm-sensors в графическом формате, осуществляющий расчёт производных показателей. Пример такой оболочки представлен на рисунке 3.

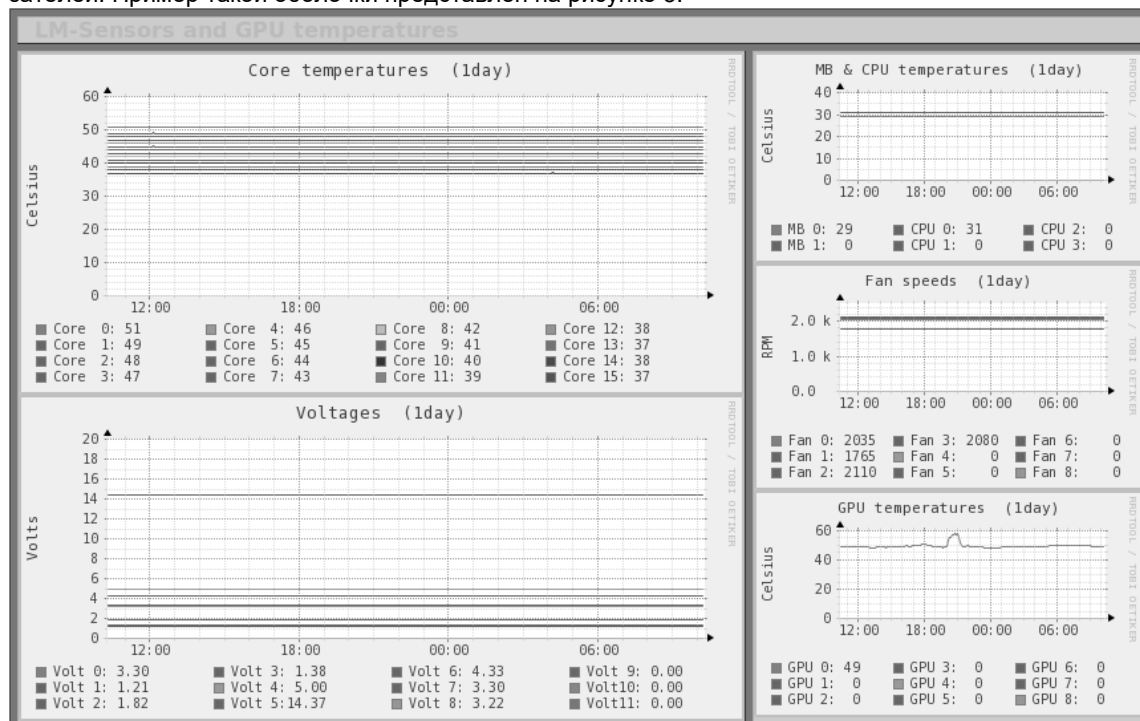


Рисунок 3 – Интерфейс утилиты Monitorix.

Еще одной распространенной утилитой является Open Hardware Monitor [Open hardware monitor documentation]. В отличие от lm-sensors, она позволяет непосредственно получать данные о потреблении энергии каждым ядром центрального процессора и процессором в целом (рисунок 4). Данное приложение исключает взаимодействие с ядром ОС, реализуя собственные методы и функции получения данных.

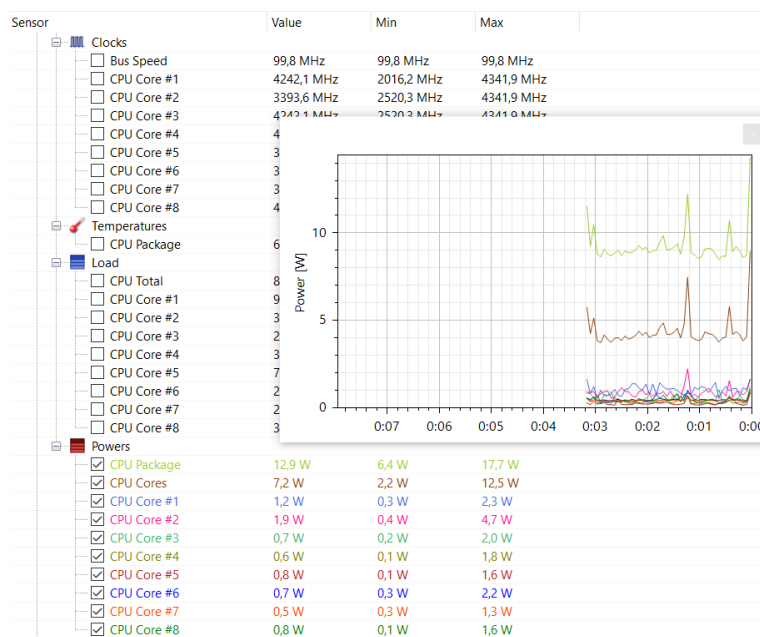


Рисунок 4. Интерфейс Open Hardware Monitor

При появлении нового чипа обновление ядра не позволяет получить метрики нового устройства и возникает проблема обновления пакета OpenHardwareMonitor. Согласно информации из открытых электронных источников, для данной программы не разрабатывались обновления с 2020 года, что определило сложности поддержки архитектур, внедрённых в указанный период.

Достоинством программы является возможность получения метрик процессора, необходимых пользователю (в частности, энергопотребления отдельных ядер). Основной недостаток состоит в том, что программа написана на языке высокого уровня C# и в операционных системах семейства Linux работает не в нативном режиме, а используя Mono, что приводит к снижению скорости работы.

Таблица 1 – Сравнение представленных программных компонентов

Программное решение	Источник данных	Отображаемые данные	Представление	
			Графическое	Текстовое
top/htop	Ядро Linux	Нагрузка процессора и процессы	-	+
lm-sensors	Модули ядра Linux	Энергопотребление и температуры	-	+
monitorix	Модули ядра Linux	Энергопотребление и температуры	+	+
Open HardwareMonitor	Системные библиотеки C#	Энергопотребление и температуры	+	+

### RAPL – Running Average Power Limit

Часто возникает необходимость оценить энергопотребление вычислительной системы в процессе выполнения проектируемой программы, возможно, различных участков проектируемой программы. Описанные выше средства могут быть использованы лишь в некоторой степени, т.к. либо не имеют API для взаимодействия с другим программным обеспечением, либо не позволяют оценить энергопотребление отдельных ядер многоядерного процессора и других элементов вычислительной системы.

Решением здесь может стать Running Average Power Limit (RAPL). RAPL представляет собой программный интерфейс, разработанный компанией Intel, который предоставляет разработчикам программного обеспечения счетчики энергопотребления микропроцессора и некоторых подсистем вычислительной системы. RAPL не является аналоговым инструментом, в нем используется оригинальная программная модель, которая для оценки энергопотребления использует встроенные счетчики производительности. Точность этих измерений зависит от типа микропроцессора, однако, разработчики утверждают, что в современных процессорах измерения осуществляются достаточно точно. Стоит также отметить, что RAPL доступен только в системах с процессорами, поддерживающими данную технологию. В первую очередь она поддерживается процессорами Intel, AMD также предоставляет совместимый интерфейс для некоторых своих систем.

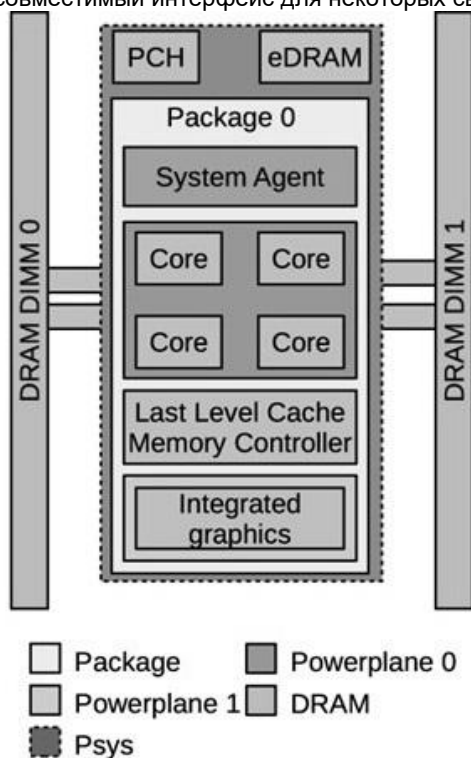


Рисунок 3. Структура доменов, поддерживаемых RAPL

Для получения данных об энергопотреблении с использованием RAPL используются регистры MSR (Model-Specific Register). Это специальные регистры, назначение которых на разных моделях процессоров может быть различным (Model-Specific – модель-зависимые). Для чтения и записи этих регистров используются привилегированные команды RDMSR и WRMSR. В состав этих регистров входят регистры для подсчета времени, мониторинга производительности, контроля отладки и других специфических функций. Доступ к привилегированным командам из прикладных программ невозможен, поэтому операционные системы предоставляют механизмы до-

RAPL позволяет осуществлять измерение энергопотребления в разных доменах (domains): собственно, сам микропроцессор, DRAM и встроенный GPU. Данная возможность появилась, начиная с архитектуры Sandy Bridge, далее развивалась во всех последующих поколениях микропроцессоров Intel. На рисунке 5 представлена структура доменов, поддерживаемых RAPL.

RAPL поддерживает следующие домены для измерения и ограничения:

- Package (PKG) – энергопотребление всего сокета;
- PowerPlane0 (PP0) – энергопотребление ядер сокета;
- PowerPlane1 (PP1) – энергопотребление GPU

(встроенная графика);

- DRAM – энергопотребление DRAM, подключенной к встроенному контроллеру памяти;

- Psys – новый домен, представленный в Intel Skylake, относящийся к энергопотреблению однокристальной системы (SOC – System-on-a-Chip); согласно рисунку 5, включает в себя в том числе потребление PCH (Platform Controller Hub – устройство, выполняющее функции южного и часть функций северного моста), eDRAM (embedded DRAM – встраиваемая DRAM).

Необходимо отметить, что не все микропроцессоры поддерживают весь набор доменов. Например, в распоряжении авторов находятся микропроцессоры Intel Xeon E5-2650 v3 2.3 GHz, входящие в состав нескольких узлов вычислительного кластера Волгоградского государственного технического университета. Данные процессоры поддерживают только два домена: PKG и DRAM.

ступа к данным регистрам. Например, входящий в состав Linux MSR Kernel Module создает псевдо-файл `"/dev/cpu/x/msr"` (с уникальным `x` для каждого процессора). Для использования данного API пользователь должен иметь разрешения на чтение содержимого данных файлов.

Авторами были проведены вычислительные эксперименты с использованием RAPL, связанные с анализом энергоэффективности программного обеспечения, реализующего базовые операции линейной алгебры, более подробно в [Kravets, 2022].

### Выводы

В настоящее время повышение энергоэффективности любого вида деятельности, будь то промышленное производство или научные исследования, инженерные расчеты или использование бытовой техники, становится как никогда актуальным. Все это в равной степени относится к IT-технологиям и разработке энергоэффективного программного обеспечения.

В работе были исследованы методы и средства контроля и повышения энергоэффективности программного обеспечения. Во-первых, были рассмотрены средства мониторинга потребления микропроцессором энергетических ресурсов, исследованы области применения и ограничения данных приложений. Во-вторых, был рассмотрен интерфейс RAPL, разработанный компанией Intel для контроля и ограничения энергопотребления микропроцессора и некоторых подсистем вычислительной системы. Данный интерфейс может использоваться для оценки энергопотребления вычислительной системы в процессе выполнения проектируемого программного обеспечения, повышая его энергоэффективность.

### Литература

1. Bán, D., Ferenc, R., Siket, I. et al. Prediction models for performance, power, and energy efficiency of software executed on heterogeneous hardware. *J Supercomput* 75, 4001–4025 (2019). <https://doi.org/10.1007/s11227-018-2252-6>
2. Fallis, E.; Spachos, P. Power Consumption and Throughput of Wireless Communication Technologies for Smartphones. In *Proceedings of the 2018 Global Information Infrastructure and Networking Symposium (GIIS)*, Thessaloniki, Greece, 23–25 October 2018 ; pp. 1–4. <https://doi.org/10.1109/giis.2018.8635665>.
3. Guzman, V., Pitkanen, T. & Takala, J. Use of compiler optimization of software bypassing as a method to improve energy efficiency of exposed data path architectures. *J Embedded Systems* 2013, 9 (2013). <https://doi.org/10.1186/1687-3963-2013-9>
4. Ilyas, A., Alatawi, M.N., Hamid, Y. et al. Software architecture for pervasive critical health monitoring system using fog computing. *J Cloud Comp* 11, 84 (2022). <https://doi.org/10.1186/s13677-022-00371-w>
5. Ivannikov, V.P., Grushin, D.A., Kuzyurin, N.N. et al. Software for improving the energy efficiency of a computer cluster. *Program Comput Soft* 36, 327–336 (2010). <https://doi.org/10.1134/S0361768810060022>
6. Katal, A., Dahiya, S. & Choudhury, T. Energy efficiency in cloud computing data centers: a survey on software technologies. *Cluster Comput* (2022). <https://doi.org/10.1007/s10586-022-03713-0>
7. Kim, D., Hong, J.E. & Chung, L. Investigating relationships between functional coupling and the energy efficiency of embedded software. *Software Qual J* 26, 491–519 (2018). <https://doi.org/10.1007/s11219-016-9346-2>
8. Kiselev, E.A., Kiselev, V.I., Shabanov, B.M. et al. The Energy Efficiency Evaluating Method Determining Energy Consumption of the Parallel Program According to Its Profile. *Lobachevskii J Math* 41, 2542–2551 (2020). <https://doi.org/10.1134/S1995080220120161>
9. Kravets, A.G., Egunov, V. The Software Cache Optimization-Based Method for Decreasing Energy Consumption of Computational Clusters. *Energies* 2022, 15, 7509. <https://doi.org/10.3390/en15207509>
10. Lm\_sensors (Linux monitoring sensors) [Электронный ресурс] // ArchWiki URL: [https://wiki.archlinux.org/title/lm\\_sensors](https://wiki.archlinux.org/title/lm_sensors) (дата обращения: 12.02.2023)
11. Lutui, P.R.; Cusack, B.; Maeakafa, G. Energy efficiency for IoT devices in home environments. In *Proceedings of the 2018 IEEE International Conference on Environmental Engineering (EE)*, Milan, Italy, 12–14 March 2018; pp. 1–6. <https://doi.org/10.1109/ee1.2018.8385277>.
12. Lyakhomskii, A.V., Perfil'eva, E.N., Kychkin, A.V. et al. A software-hardware system of remote monitoring and analysis of the energy data. *Russ. Electr. Engin.* 86, 314–319 (2015). <https://doi.org/10.3103/S1068371215060103>
13. Majeed, A., Shah, M.A. Energy efficiency in big data complex systems: a comprehensive survey of modern energy saving techniques. *Complex Adapt Syst Model* 3, 6 (2015). <https://doi.org/10.1186/s40294-015-0012-5>
14. Noureddine, A., Rouvoy, R. & Seinturier, L. Monitoring energy hotspots in software. *Autom Softw Eng* 22, 291–332 (2015). <https://doi.org/10.1007/s10515-014-0171-1>
15. Open hardware monitor documentation [Электронный ресурс] URL: <https://openhwaremonitor.org/documentation/> (дата обращения: 06.02.2023)
16. Schill, A.; Globa, L.; Stepurin, O.; Gvozdetzka, N.; Prokopets, V. Power Consumption and Performance Balance (PCPB) scheduling algorithm for computer cluster. In *Proceedings of the 2017 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*, Odessa, Ukraine, 11–15 September 2017; pp. 1–8. <https://doi.org/10.1109/ukrmico.2017.8095365>.
17. The Linux kernel user-space API guide [Электронный ресурс] // The Linux Kernel URL: <https://docs.kernel.org/userspace-api/index.html> (дата обращения: 12.02.2023)

*Егунув Виталий Алексеевич – доцент кафедры «ЭВМ и системы» Волгоградского государственного технического университета, Волгоград, Россия,  
ORCID: 0000-0001-9087-3275, vegunov@mail.com*

*Абдрахманов Дмитрий Леватович – аспирант кафедры «Физика» Волгоградского государственного технического университета, Волгоград, Россия,  
565963@gmail.com*

**Ключевые слова:** энергоэффективность программного обеспечения, оптимизация, мониторинг, вычислительный кластер, RAPL, Im-sensors, Linux, UNIX

**Egunov Vitaly Alekseevich, Abdrahmanov Dmitry Levatovich. Analysis of Means of Control and Improvement of Software Energy Efficiency**

**Keywords:** software energy efficiency, optimization, monitoring, computing cluster, RAPL, Im-sensors, Linux, UNIX

Egunov Vitaly Alekseevich – Associate Professor, Computers and Systems Department, Volgograd State Technical University, Volgograd, Russia,  
ORCID: 0000-0001-9087-3275, vegunov@mail.com

Abdrahmanov Dmitry Levatovich – Postgraduate Student, Physics Department, Volgograd State Technical University, Volgograd, Russia,  
565963@gmail.com

DOI: 10.34706/DE-2023-02-08

JEL classification L86 – Информация и интернет-сервисы, компьютерное программное обеспечение

Currently, improving the energy efficiency of software and computing systems in general is becoming more relevant than ever. Methods and means of control and improvement of energy efficiency of software were investigated in the work. The means of monitoring the consumption of energy resources by microprocessors were considered, the application areas and limitations of these applications were investigated. The API interface developed by Intel to control and limit the power consumption of microprocessors and some subsystems of the computing system was also considered.